DOI: 10.14421/ijid.2023.4308 IJID (International Journal on Informatics for Development), *e-ISSN: 2549-7448*
Vol. 12, No. 2, December 2023, Pp. 374-385

Anomaly-Based Intrusion Detection System for the Internet of Medical Things

Eichie Franklin
Department of Engineering and Mathematics
Sheffield Hallam University
Sheffield, UK

Bernardi Pranggono
School of Computing and Information Science
Anglia Ruskin University
Cambridge, UK
-

Article History
Received December 13th, 2023
Revised March 16th, 2024
Accepted March 29th, 2024
Published April, 2024

***Abstract*— The use of the Internet of Things (IoT) in the health sector, known as the Internet of Medical Things (IoMT), allows for personalized and convenient (e)-health services for patients. However, there are concerns about security and privacy as unethical hackers can compromise these network systems with malware. We proposed using hyperparameter-optimized Machine and Deep Learning models to address these concerns to build more robust security solutions. We used a representative Anomaly Intrusion Detection System (AIDS) dataset to train six state-of-the-art Machine Learning (ML) and Deep Learning (DL) architectures, with the Synthetic Minority Oversampling Technique (SMOTE) algorithm used to handle class imbalance in the training dataset. Our hyperparameter optimization using the Random search algorithm accurately classified normal cases for all six models, with Random Forest (RF) and K-Nearest Neighbors (KNN) performing the best in accuracy. The attention-based Convolutional Neural Network and Long Short-Term Memory (CNN-LSTM) model was the second-best performer, while the hybrid CNN-LSTM model performed the worst. However, there was no single best model in classifying all attack labels, as each model performed differently in terms of different metrics.**

Keywords—anomaly detection; e-health; internet of things; malware; unethical hackers

1 INTRODUCTION

Internet of Things (IoT) transformation studies have long been extended to the health sector and are mainly referred to as the Internet of Medical Things (IoMT) [1], [2], [3], [4]. IoMT makes personalized (e)-health services possible and enjoyable for patients who use them. Several benefits of IoMT among others according to Liyakathunisa et al. include - infectious disease remote monitoring; effective treatment and diagnosis; constant monitoring of patients' conditions; instant access to the patient's medical history; fast notification and automatic reminders; remote medical care; automatic transmission and analysis of data generated from IoMT devices; ease of embedding advanced and accurate algorithms that can detect abnormalities; ease of locating and tracking patients; and flexibility of having medical consultations remotely via telehealth and telemedicine [5].

The paper aims to contribute to the development of an effective anomaly-based Intrusion Detection System (AIDS) [6], [7], [8] for IoMT by revisiting the implementation of various machine learning (ML) and deep learning (DL) algorithms on a recent and relevant realistic public dataset. Due to its benefits and advantages, the adoption of IoT devices in healthcare organizations has reached 70% with increasing reliance in such organizations on the IoMT. The COVID-19 pandemic increased the adoption of the IoMT to reduce the risks of getting infected while treating patients. It is expected that the global IoT in the healthcare market will reach USD 290 billion by 2028 from USD 128 billion in 2023. However, we also see increasing cyber-attacks during the pandemic where cyber criminals and Advanced Persistent Threat (APT) groups have taken advantage of targeting vulnerable people and systems [9]. The increase in connectivity in IoMT also creates an increase in the risk of security breaches and cyber-attacks. Hackers may target IoMT devices to access sensitive patient data or disrupt critical medical processes. Therefore, it is essential to develop effective intrusion detection systems (IDSs) to protect the IoMT from cyber threats [10], [11].

The contributions of the paper are as follows:

- Examining the existing intrusion detection systems designed for IoMT.
- Investigating and evaluating the use of six machine learning and deep learning algorithms: Random Forest (RF), Support Vector Classifier (SVC), K-Nearest Neighbor (KNN), Convolutional Neural Network (CNN), hybrid Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) (CNN-LSTM), and the attention-based hybrid CNN-LSTM with hyperparameter-optimized AIDS for IoMT.
- Implementing an intrusion detection system for IoMT with a realistic dataset.
- Suggesting future work in the field.

Due to the widespread cyberattacks on IoMT, various intrusion detection systems (IDSs) for IoMT have been developed in recent years. In this section, we highlight some of these studies.

Binbusayyis et al. proposed an investigation and comparison platform to understand the efficiency of the ML algorithm for intrusion alert in the IoMT network [12]. Five ML algorithms including the KNN, Naïve Bayes (NB), Support Vector Machine (SVM), Artificial Neural Network (ANN), and Decision Tree (DT) algorithms are investigated over the publicly available Bot-IoT dataset. The study reported that the DT algorithm outperformed other ML algorithms in detecting intrusion.

Thamilarasu et al. applied ML techniques and mobile agent technology to design and develop an attack detection system connecting several medical IoT devices [13]. Their system used three agents whose specific task was to migrate, learn and collaboratively perform attack detection. Using various ML algorithms such as the popular SVM, DT, NB, KNN, and RF, the study performed experiments with several wireless networks connected to medical IoT devices. Results obtained in the work demonstrated high detection accuracy with minimal energy consumption overhead.

Zachos et al. proposed a hybrid system architecture for anomaly detection in IoMT networks [14]. They leveraged host-based and network-based technologies to monitor and collect log files from the IoMT devices, the gateway, and the traffic from the IoMT edge network. Their anomaly detection system can minimize the computational cost using ML techniques, which are implemented by a detection engine running on the gateway of the IoMT edge network. Popular ML algorithms such as the KNN, NB, DT, and RF algorithms are built and implemented over two current IoT datasets. Results obtained from the experiment suggest that DT, RF, and KNN algorithms are most suitable for the Central Detection component of the proposed AIDS.

A method to detect attack traffic using a deep neural network in the IoMT-Blockchain environment is proposed in [15]. The authors employed a multi-model autoencoder (MMAE) to effectively learn the fusion of low-dimensional feature representations from various characteristics of the original data. The study used two proprietary datasets (TADA and TADB) gathered from the IoMT-Blockchain network. TADA included DoS, Probe, R2L, PortScan, SSH, and U2R, while TADB included Backdoor, DoS, Exploit, Analysis, Fuzers, and Worms.

Anomaly-based IDS in IoT using kernel extreme learning machine to classify malicious traffic is proposed by [16]. The study showed that the proposed method can improve the performance of IDS in terms of accuracy rate, sensitivity rate, F1-score and the area under the curve.

From the above-reviewed literature, it is evident that the use of ML and DL techniques with hyperparameter optimization tuning and data augmentation that can surmount the challenges of conventional machine learning models and result in better performance for similar applications has not been fully explored. Hence, we propose to leverage this advantage to advance research in IoMT AIDS modelling. In this study, we built an AIDS for IoMT network that would leverage the hyperparameter optimization of ML and DL algorithm parameters and evaluate their performance. We used six ML and DL algorithms: RF, SVC, KNN, CNN, CNN-LSTM, and the attention-based hybrid CNN-LSTM.

We also used the popular realistic public dataset suitable for IoMT in our experiments.

2 METHOD

2.1 Dataset

The use of IoT-related datasets that reflect real-world IoT applications plays an essential role in evaluating the accuracy as well as the efficiency of the intrusion detection models. However, there is a lack of availability of real-world datasets among the research community as most of the companies that deal with IoT devices are reluctant to share their log details due to privacy concerns. This creates an obstacle in the creation of intrusion detection models tailored to IoT, IoMT, or Industrial IoT (IIoT) applications.

We collected the publicly available and best dataset for AIDS applications that is representative of the current attacks on IoMT devices and networks. Specifically, we used the TON_IoT telemetry dataset [17], [18], [19]. The TON_IoT datasets are new generations of industry 4.0/ IoT and IIoT datasets for evaluating the fidelity and efficiency of different cybersecurity applications based on AI, that is, ML and DL algorithms. We labelled all attacks and normal traffic individually using class 0 to represent the normal cases, and classes 1–7 to represent various attacks. These attacks include a backdoor attack represented by class 1, an Injection attack represented by class 2, a password attack represented by class 3, a DDoS attack represented by class 4, a ransomware attack represented by class 5, an XSS attack represented by class 6, and scanning attack represented by class 7. The distribution of the normal and attacks consists of 210,000 normal cases data points, 30,000 backdoor attacks data points, 30,000 injection attacks data points, 30,000 password attacks data points, 20,000 DDoS attacks, 13,128 ransomware attacks, 4,960 XSS attacks, and 3,444 scanning attacks.

2.2 Data Pre-Processing

We pre-processed our data by making x and y variables from it. This was done by setting the label column as y target variable and encoding it. We also set other columns excluding the label and type columns as x variables. From Figure 1, we can see that the study dataset is biased. This is known as a problem of class imbalance, and it occurs when there is an unequal distribution of resources among classes. Consequently, we must perform data augmentation on it to remove bias and create equal distributions. To accomplish this, we used the Synthetic Minority Oversampling Technique (SMOTE) algorithm [20], [21], [22]. By default, SMOTE is designed to oversample all classes to have the same number of examples as the class with the highest number of examples. In our case, class 0, which represents the normal cases, has the highest number of examples with 210,000 data points. Therefore, the SMOTE algorithm will oversample all classes to have 210,000 examples. This can be seen in Figure 1 that shows the study data distribution after the data augmentation process.

The SMOTE technique was applied only to the training data to address the issue of class imbalance. By oversampling the minority class in the training data, we aimed to provide a more balanced view of the model, thereby improving its ability to detect anomalies, which are often underrepresented.

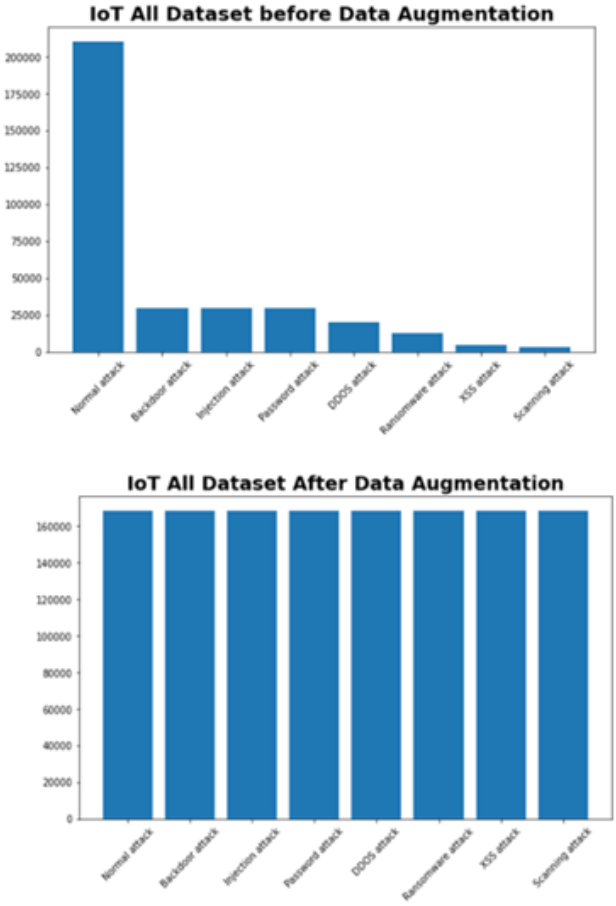

Figure 1. The distribution of the training dataset before and after data augmentation

Applying SMOTE to the entire dataset, including the test data, could lead to overly optimistic performance estimates. The reason is that it could result in having identical instances in both the training and test sets. This would give an illusion that the model is performing well on unseen data, while in reality, it has already seen those instances during training. By applying SMOTE only to the training data, we ensured that the test data served as a realistic representation of the original data distribution and provided a reliable estimate of the model performance. We also normalized our *x* variable training and testing dataset to scale it into the range of 0 and 1 using the MinMaxScaler function [23].

2.3 Machine Learning Models

Three ML models including the RF, the SVM, and the KNN classifier models were compiled and built. The aim was to use the ML classifiers to fit our IoMT training dataset by supervised learning and make predictions classifying the

labels into normal cases and cyber-attack cases. These algorithms are briefly described in the following subsection.

2.3.1 *RF Algorithm:* The RF classifier algorithm is compiled and built in this study using the Scikit-Learn ML library. The RF algorithm is a robust and popular ensemble algorithm for classification tasks *[24]*. It is known for its scalability and ease of use.

2.3.2 *SVM Algorithm: We used the* Scikit-Learn ML library to compile and build the SVM classifier method. Its efficiency as an ML algorithm has led to its use in a wide range of fields, including the categorization of electrocardiograms. A more involved version of the SVC was utilized by Jannah et al. *[25]*. The SVC is capable of both binary and multi-class classification on any given dataset. We used the Scikit-learn ML toolkit in Python to compile and create the SVC algorithm using a random state value of zero, a tolerance value of 0.00001, and other default settings.

2.3.3 *KNN Algorithm:* The KNN classifier algorithm was compiled and built in this study using the Scikit-Learn ML library. KNN is a straightforward algorithm that is a bit different from the SVC classifiers described earlier. The difference is seen in their learning process. For example, the KNN algorithm is a non-parametric type of ML model whose learning process is instance-based. That is, their learning is characterized by memorizing the training set resulting in no (zero) cost. On the other hand, the models cannot be characterized by a fixed set of parameters and the number of parameters is a function of the size of the training set. In contrast, the linear SVC classifier is a parametric type of ML model that allows the estimation of parameters from the training set to learn a function that can classify new data points.

2.4 Deep Learning (DL) Models

Three DL models including the CNN, the CNN-LSTM, and the Attention-CNN-LSTM classifier models were compiled and built. The aim was to use the DL classifiers to fit our IoMT training dataset by supervised learning and make predictions classifying the labels into normal cases and cyber-attack cases. We used the Random Search hyperparameters optimization technique [26] to find the best parameters for compiling and building an efficient DL model. Following that, the best hyperparameters were obtained and utilized to train DL models on the training data.

2.4.1 *CNN Classifier:* CNN has grown in popularity in a variety of AI applications such as image classification, speech recognition, computer vision, etc. *[27]*. The network comprises layers that extract low-level features from raw data and use other layers to process these features for an output. We used tabular data as input for all models including the CNN model. Each row in the table represents a network event, and each column represents a feature of the network event, such as packet size, time, protocol type, type of traffic, etc. The data were not spatially correlated as in images, but there might be some form of temporal correlation between different network events. To apply CNNs, traditionally used for image data, to our tabular data, we made some adjustments to the model architecture. Instead of treating the input as a 2D image, we treated it as a 1D sequence. Our CNN model was designed with 1D convolutional layers instead of the typical 2D convolutional layers used for image data. The 1D convolutional layers can capture the local dependencies of the input sequence, which can be crucial for identifying anomalous patterns in network traffic. The output of the convolutional layers was then flattened and fed into fully connected layers for the final classification. The 1D CNN architecture followed the Keras sequential modelling style that first defined the model before adding other layers. Three blocks of the basic structure of a CNN were included in this architecture (see Figure 2). Block A consisted of a 1D convolutional layer with 128 filters, kernel size of 6, ReLU activation function, ‘same’ padding, and input shape of (13, 1); a Batch Normalization layer; and a Max Pooling 1D layer with pool size of 3, strides of 2, and ‘same’ padding. Block B consists of a 1D convolutional layer with 64 filters, kernel size of 6, ReLU activation function, ‘same’ padding, and input shape of (13, 1); a Batch Normalization layer; and a Max Pooling 1D layer with pool size of 3, strides of 2, and ‘same’ padding. Block C consisted of a 1D convolutional layer with 64 filters, kernel size of 6, Rectified Linear Unit (ReLU) activation function, ‘same’ padding, and input shape of (13, 1); a Batch Normalization layer; and a Max Pooling 1D layer with pool size of 3, strides of 2, and ‘same’ padding. The output from the three blocks was flattened with the addition of a flattened layer. A dropout of 20% was added and the output was passed to a dense layer with 64 neurons in the presence of a ReLU activation function. Another dropout of 20% was added and the output was passed to another dense layer with 64 neurons in the presence of a ReLU activation function. Again, another dropout of 20% was added and the output was passed to a dense layer with 8 neurons in the presence of a SoftMax activation function. The SoftMax activation function was added to compute the probabilities for each target class in the total classes. Furthermore, the Stochastic Gradient Descent (SGD) optimizer was used in this study with the categorical cross-entropy loss function. An early stopping criterion with a patience of 10 was also implemented. Total parameters of 88,712, 88,200 trainable and 512 non-trainable,

were included in this architecture. Figure 2 shows the CNN architecture built in this study.

2.4.2 *Hybrid Convolutional Neural Network and Long Short-Term Classifier:* Recurrent neural network (RNN) algorithms can overcome the traditional NN's limitation in extracting sequence data-based information [27], [28]. The algorithms use network types that are related to previous outputs, making modelling time series problems possible. However, they become very inefficient when presented with larger sequences. The LSTM algorithm uses cell state memory instead of simple neurons which helps the algorithm to store data for longer periods. LSTMs have a repeating module with four interacting layers. Its building block is the memory cell. In each memory cell, there is a recurrent edge that has a desirable weight of w=1 to overcome the vanishing and exploding gradient challenges. The values associated with this recurrent edge are called the cell state. The network structure consists primarily of the cell state and gates, which include the input gate, forget gate, and output gate, and these gates can show or erase data in the cell state memory block at random time intervals [28]. The major function of the cell state is to set the current state of the cell in the algorithm among others. In this study, we compiled and built a hybrid CNN-LSTM model. The CNN architecture compiled here is like the previous one described earlier. The LSTM layer was added to the CNN architecture to form the hybrid CNN-LSTM model. The LSTM layer was compiled with 256 hidden neurons, 20% dropout, and 20% recurrent dropout. This layer was added to the CNN architecture after the block C layer to form the hybrid CNN-LSTM model as seen in Figure 3. Additionally, only one dense layer of 64 neurons was added here contrary to that of the CNN architecture. The last dense layer consists of 8 hidden neurons and a SoftMax function. The SGD optimizer was used with the categorical cross-entropy loss function. An early stopping criterion with a patience of 10 was also implemented. Total parameters of 421,448, 420,936 trainable and 512 non-trainable, were included in this architecture.

2.4.3 *Attention-based Convolutional Neural Network and Long Short-Term Classifier:* In 2016, Bahdanau et al. proposed the widely used attention process model as a component of a neural network and adapted it to neural-machine translation [29]. Two popular variants of these models exist, and they are the Bahdanau model [29] and the Luong model [30]. In this study, we compiled and built an attention-based hybrid CNN-LSTM model. The architecture consisted of the hybrid CNN-LSTM model and an attention layer. The attention layer was added immediately after the LSTM layer of the hybrid CNN-LSTM model as seen in Figure 4. Other parameters were similar to the hybrid CNN-LSTM model. The SGD optimizer was used with the categorical cross-entropy loss function. An early stopping criterion with a patience of 10 was also implemented. Total parameters of 421,706—421,194 trainable and 512 non-trainable—were included in this architecture.

2.5 *Train and Test Model*

The six compiled and built ML and DL models were trained using the training set and the best-optimized parameters obtained from the random search optimization operation. However, it is important to state that the ML algorithms were not optimized in this study since they did not need to use the learning rate parameter like the DL algorithms. The DL algorithms are trained for 10 epochs on a computer with processor Intel (R) Xeon (R) CPU E30-1246v3@ 350 GHz, 16.0GB, 64-bit operating system, X64-based processor, with Windows 10 Education. The computing resource is provided by Google Colab with GPU access and details: NVIDIA-SMI 460.32.03 Driver version: 460.32.03 CUDA version: 11.2 Tesla T4. After training, the models are tested on the tested and evaluated on the set to obtain the classification results

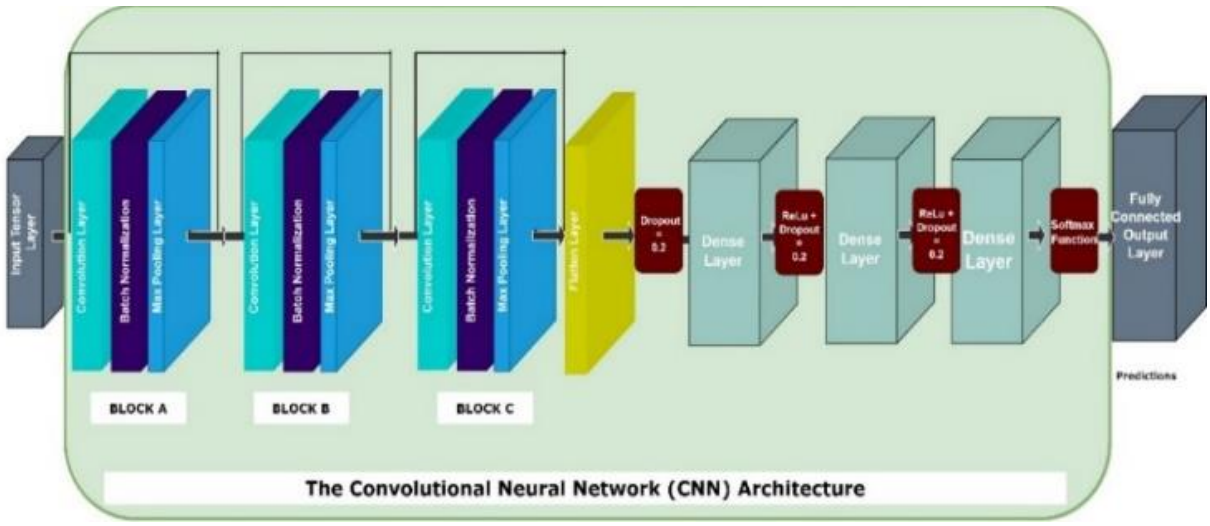

Figure 2. CNN architecture built in this study

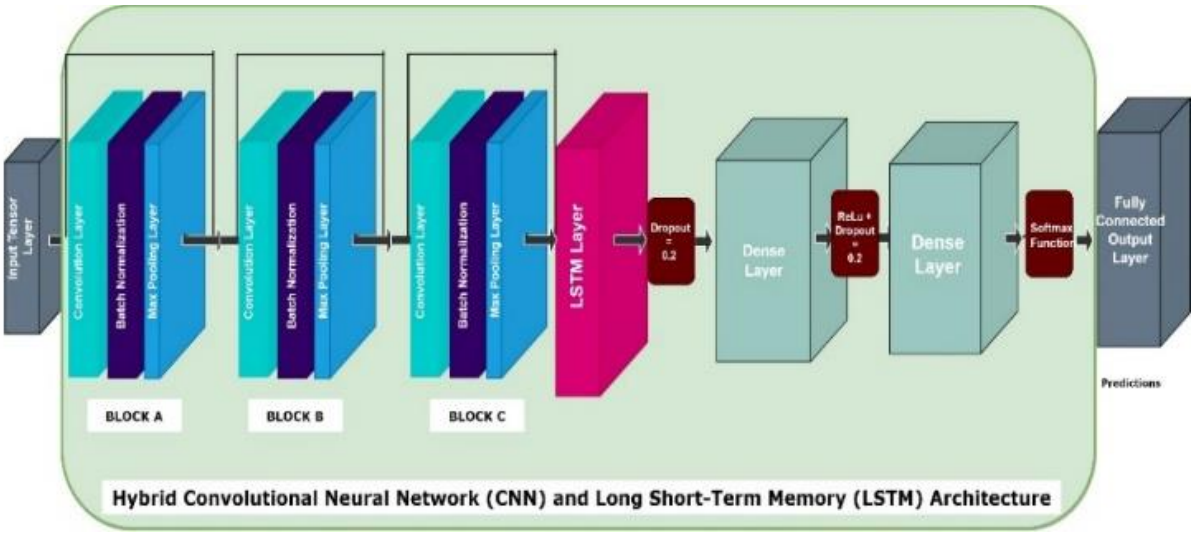

Figure 3. Hybrid CNN-LSTM model

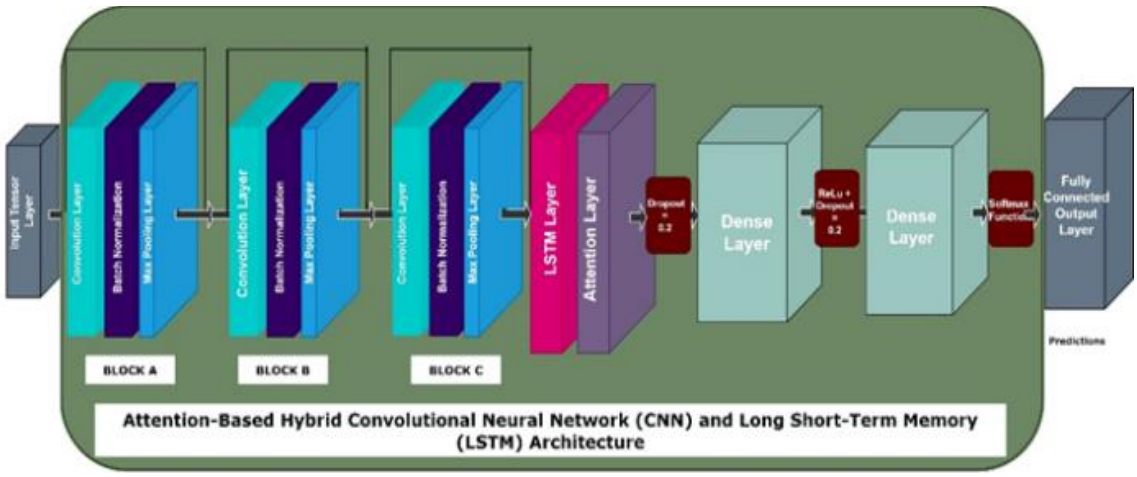

Figure 4. Attention-based Hybrid CNN-LSTM model

2.6 Performance Metrics

The ML and DL models were evaluated using standard performance metrics: accuracy, precision, recall, and F1-score (see Table 1) [7]. Where true positive (TP) means anomalous traffic correctly identified, true negative (TN) means normal traffic correctly identified, false positive (FP) means normal traffic incorrectly identified as anomalous, and false negative (FN) means anomalous traffic incorrectly identified as normal.

Table 1. Performance Metrics

Performance Metric	Definition
Accuracy	$\frac{TP + TN}{(TP + TN + FP + FN)}$
Precision	$\frac{TP}{(TP + FP)}$
Recall (True Positive rate)	$\frac{TP}{(TP + FN)}$
F1-Score	$\frac{2\ x\ Precision\ x\ Recall}{(Precision + Recall)}$

3 RESULT AND DISCUSSION

3.1 Classification Results

The classification results for six ML and DL algorithms are presented in Table 2 – Table 7. Table 2 presents the classification report of the predictions from the RF architecture. Figure 5 shows the Error matrix of the True labels versus Predicted labels. For the ML models, from Table 2 and Figure 5, it is observed that the RF model efficiently classified all labels, both normal and cyber-attacks with 100% certainty in terms of Precision, Recall, and F1-Score metrics, and with 99% Accuracy, except with the password and scanning cyber-attacks. The model classified the password attack with 97% Precision, 93% Recall, 95% F1-Score and 99% Accuracy, while it classified the scanning

attack with 58% Precision, 78% Recall, 66% F1-Score, and 99% Accuracy.

Table 3 presents the classification report of the predictions from the SVC architecture. From Table 3, it is shown that the SVC model efficiently classified the normal cases with 100% certainty in terms of Precision, Recall, and F1-Score metrics, and with 80% Accuracy. For the label attacks, the model classified the backdoor attack with 99% Precision, 97% Recall, 98% F1-Score and 80% Accuracy, while it classified the ransomware attack with 80% Precision, 93% Recall, 86% F1-Score, and 80% Accuracy. It also classified the Injection attack with 74% Precision, the XSS attack with 54% Recall, and the scanning attack with 85% Recall score. All other cyber-attack labels are poorly classified by the model.

Table 4 presents the classification report of the predictions from the KNN architecture. From Table 4, it is observed that the KNN model efficiently classified the normal cases, the DDoS attack, and the ransomware attack with 100% certainty in terms of Precision, Recall, and F1-Score metrics, and with 99% Accuracy. For other attack labels, the model classified them efficiently also with performance between 92% – 99% in terms of all metrics, except with the scanning attack where it achieved 55% Precision, 76% Recall, 64% F1-Score, and 99% Accuracy.

Table 2. RF Performance Results

Class	Precision	Recall	F1-Score	Accuracy (weighted average)
0	1.00	1.00	1.00	0.99
1	1.00	1.00	1.00	0.99
2	1.00	1.00	1.00	0.99
3	0.97	0.93	0.95	0.99
4	1.00	1.00	1.00	0.99
5	1.00	1.00	1.00	0.99
6	1.00	1.00	1.00	0.99
7	0.58	0.78	0.66	0.99

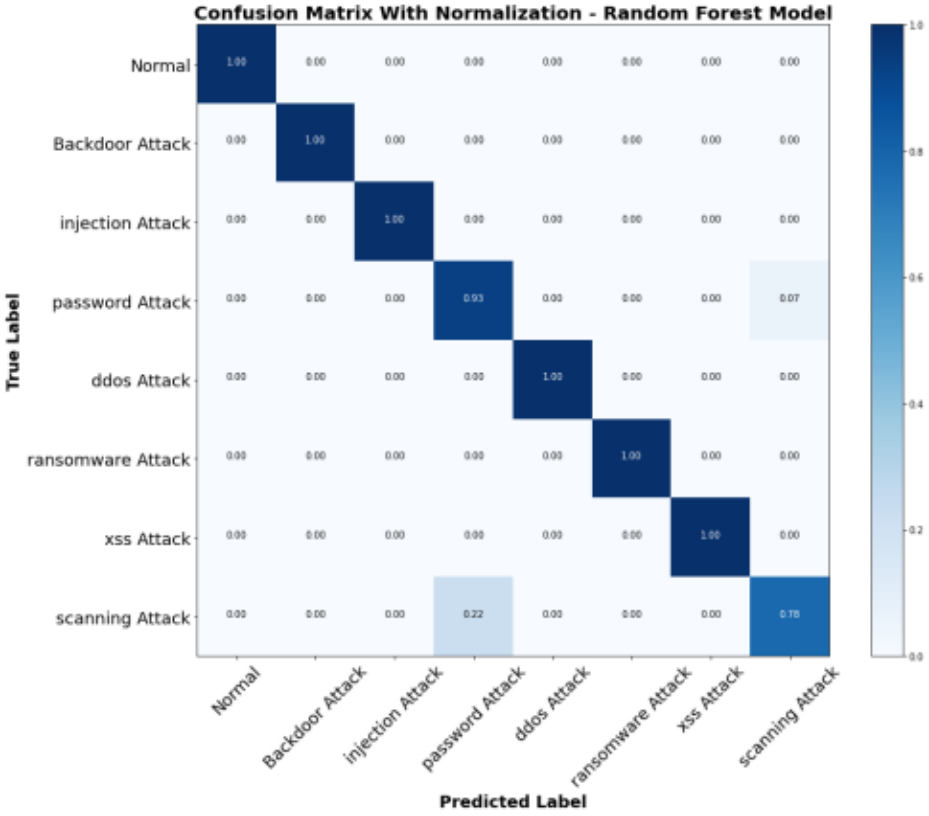

Figure 5. RF error matrix of the true label versus predicted label

Table 3. SVC Performance Results

Class	Precision	Recall	F1-Score	Accuracy (weighted average)
0	1.00	1.00	1.00	0.80
1	0.99	0.97	0.98	0.80
2	0.74	0.01	0.03	0.80
3	0.46	0.30	0.37	0.80
4	0.41	0.36	0.38	0.80
5	0.80	0.93	0.86	0.80
6	0.13	0.54	0.20	0.80
7	0.11	0.85	0.19	0.80

Table 4. KNN Performance Results

Class	Precision	Recall	F1-Score	Accuracy (weighted average)
0	1.00	1.00	1.00	0.90
1	0.81	1.00	0.90	0.90
2	0.92	0.48	0.63	0.90
3	0.79	1.00	0.88	0.90
4	0.53	0.89	0.66	0.90
5	0.95	0.51	0.66	0.90
6	0.00	0.00	0.00	0.90
7	0.00	0.00	0.00	0.90

Table 5 presents the classification report of the predictions from the CNN architecture. Figure *6*6 shows the Error matrix of the True labels versus Predicted labels. For the DL models, from Table 5 and Figure *6*6, it can be seen that the CNN model efficiently classified the normal cases with 100% certainty in terms of Precision, Recall, and F1-Score metrics, and with 90% Accuracy. For the attack labels, the model classified the backdoor attack with 81% Precision, 100% Recall, 90% F1-Score and 90% Accuracy; the injection attack with 92% Precision, 48% Recall, 63% F1-Score, and 90% Accuracy; the password attack with 79% Precision, 100% Recall, 88% F1-Score, and 90% Accuracy; the DDoS attack with 53% Precision, 89% Recall, 66% F1-Score, and 90% Accuracy; and the ransomware attack with 95% Precision, 51% Recall, 66% F1-Score, and 90% Accuracy. The model could not predict the XSS and the scanning attacks in Precision, Recall, and F1-Score, but it achieved 90% in classifying these attacks.

Table 6 presents the classification report of the predictions from the CNN-LSTM architecture. Figure *7*7 shows the Error matrix of the True labels versus Predicted labels. From Table 6 and Figure *7*7, it is evident that the hybrid CNN-LSTM model efficiently classified the normal cases with 100% certainty in terms of precision, Recall, and F1-Score metrics, and with 71% accuracy. For the attack labels, the model classified the injection attack with 45% precision, 100% Recall, 62% F1-Score, and 71% accuracy. The model poorly classified every other attack label in precision, Recall, and F1-Score, but it achieved 71% in accurately classifying these other attack labels.

Error! Reference source not found. presents the classification report of the predictions from the attention-based CNN-LSTM architecture. Figure *8*8 shows the Error matrix of the True labels versus Predicted labels. From **Error! Reference source not found.** and Figure *8*8, it is evident that the attention-based CNN-LSTM model efficiently classified the normal cases with 100% certainty in Precision, Recall, and F1-Score metrics, and with 94% Accuracy. For the attack labels, the model classified the backdoor attack with 99% Precision, 92% Recall, 95% F1-Score and 94% Accuracy; the injection attack with 94% Precision, 65% Recall, 77% F1-Score, and 94% Accuracy; the password attack with 82% Precision, 100% Recall, 90% F1-Score, and 94% Accuracy; the DDoS attack with 65% Precision, 100% Recall, 79% F1-Score, and 94% Accuracy; and the ransomware attack with 84% Precision, 99% Recall, 91% F1-Score, and 94% Accuracy. The model could not predict the XSS and the scanning attacks in precision, Recall, and F1-Score, but it achieved 94% in accurately classifying these attacks.

Table 5. CNN Performance Results

Class	Precision	Recall	F1-Score	Accuracy (weighted average)
0	1.00	1.00	1.00	0.90
1	0.81	1.00	0.90	0.90
2	0.92	0.48	0.63	0.90
3	0.79	1.00	0.88	0.90
4	0.53	0.89	0.66	0.90
5	0.95	0.51	0.66	0.90
6	0.00	0.00	0.00	0.90
7	0.00	0.00	0.00	0.90

Table 6. CNN-LSTM performance results

Class	Precision	Recall	F1-Score	Accuracy (weighted average)
0	1.00	1.00	1.00	0.71
1	0.00	0.00	0.00	0.71
2	0.45	1.00	0.62	0.71
3	0.05	0.07	0.06	0.71
4	0.07	0.08	0.07	0.71
5	0.00	0.00	0.00	0.71
6	0.00	0.00	0.00	0.71
7	0.00	0.00	0.00	0.71

Table 7. Attention-based CNN-LSTM performance results

Class	Precision	Recall	F1-Score	Accuracy (weighted average)
0	1.00	1.00	1.00	0.94
1	0.99	0.92	0.95	0.94
2	0.94	0.65	0.77	0.94
3	0.82	1.00	0.90	0.94
4	0.65	1.00	0.79	0.94
5	0.84	0.99	0.91	0.94
6	0.00	0.00	0.00	0.94
7	0.00	0.00	0.00	0.94

Generally, all six ML and DL models accurately classified the normal cases in terms of the Precision, Recall, and F1-score metrics. In terms of the Accuracy metric, the RF and the KNN models achieved the best with 99% in classifying the normal cases; the attention-based hybrid CNN-LSTM model trained for only 1 epoch achieved the second best with 94%; the CNN model trained for only 1 epoch achieved the third best with 90%; the SVC model achieved the 4th best with 80%; and the hybrid CNN-LSTM model was the worst performed with 71%. In terms of other metrics and attack labels, different models performed differently. However, though we could only train the DL models for only 1 epoch due to computational resources, it is evident that the RF and the KNN models are best performed in terms of all metrics. The attention-based hybrid CNN-LSTM model is second best, while the hybrid CNN-LSTM model performed the worst.

3.2 Discussion

In our study, we observed variations in the performance of our model across different attack labels. This could be attributed to the inherent differences in the characteristics of different types of attacks. For instance, some attacks may have distinct patterns that are easier for the model to learn and identify, resulting in higher performance metrics for these attack labels. On the other hand, some attacks may exhibit patterns that are more subtle or similar to normal traffic, making them harder to detect and leading to lower performance metrics.

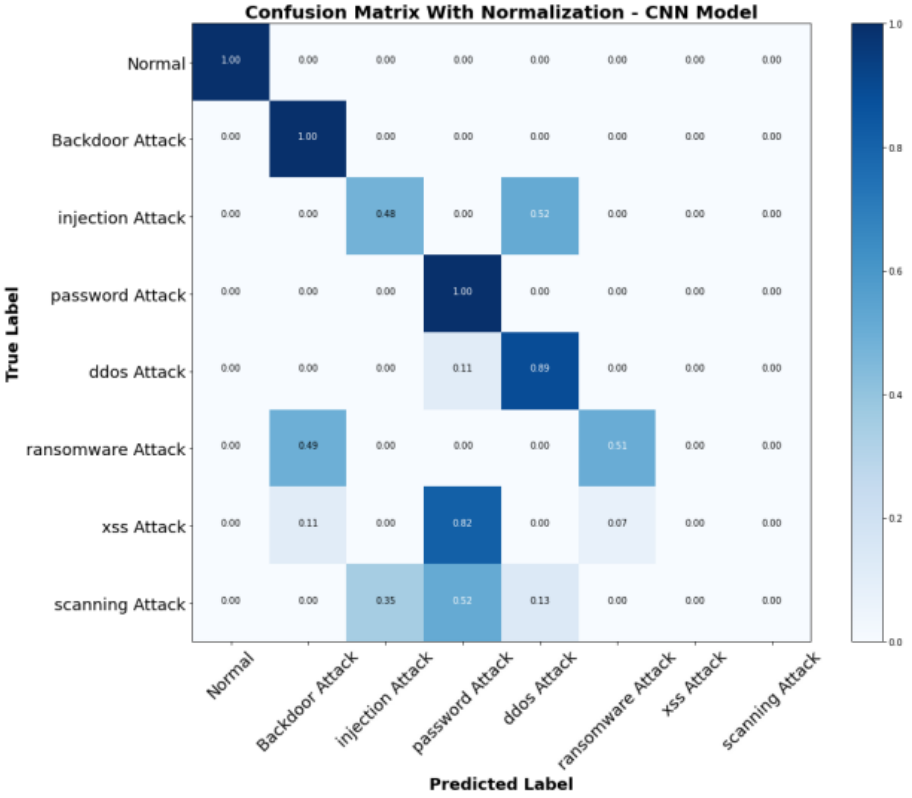

Figure 6. CNN error matrix of the true label versus predicted label

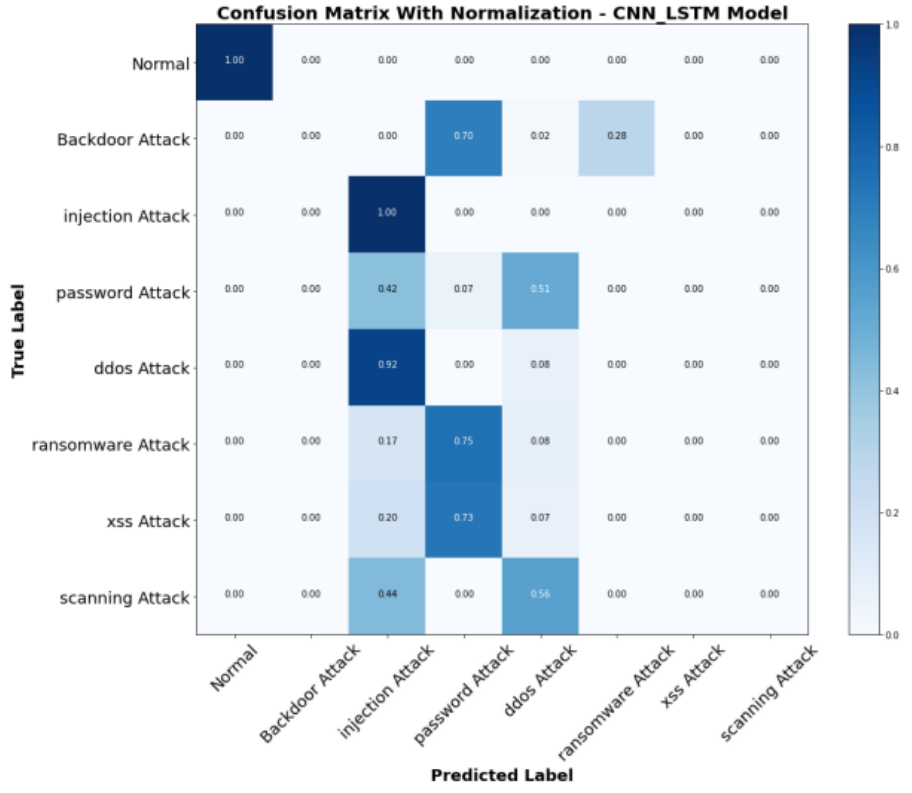

Figure 7. CNN-LSTM error matrix of the true label versus predicted label

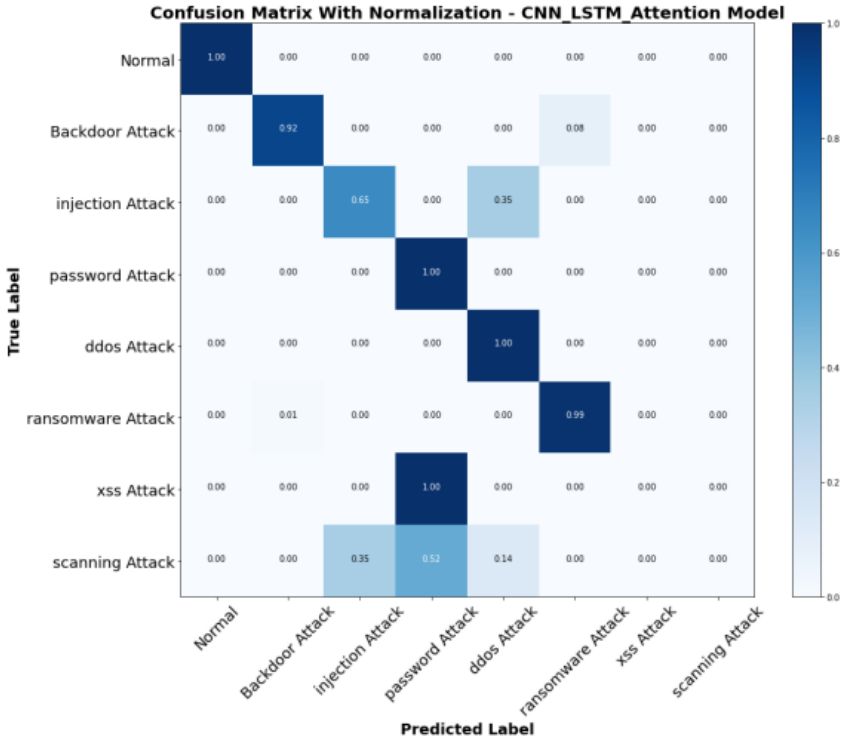

Figure 8. Attention-based CNN-LSTM error matrix of the true label versus predicted label

The distribution of different attack labels in our dataset could also influence the variations. If certain attack labels are underrepresented in the dataset, the model may not have sufficient examples to learn from, affecting its ability to accurately identify these attacks. These variations in model performance highlight the importance of having a diverse and representative dataset for training. In real-world applications, an intrusion detection system may encounter a wide range of attacks, some of which may not be well-represented in the training data. This underscores the need for continuous model training and updating as new attack patterns emerge.

The dataset used in our study was collected from a specific IoMT environment, which may not fully represent the diversity of IoMT devices and network traffic patterns in other environments. This could introduce biases in our model, as it might perform well on similar data but fail to generalize to different IoMT environments. Furthermore, any inherent biases or errors in the dataset could have been inadvertently learned by our model, affecting its performance and reliability.

Future studies could benefit from incorporating multiple datasets collected from diverse IoMT environments. This would provide a more comprehensive and representative view of IoMT network traffic, enhancing the generalizability of the anomaly detection system. Additionally, using multiple datasets could help identify and mitigate potential biases or errors in individual datasets. The significance of larger training epochs for DL architectures lies in the potential for improved model performance. With more epochs, the model has more opportunities to learn from the data and adjust its weights and biases to minimize the loss function. However, it is important to monitor the model for signs of overfitting, as training for too many epochs can cause the model to become overly specialized to the training data and perform poorly on unseen data.

Future studies could also consider a wider range of IoMT-related features in the training set. For instance, in addition to network traffic features, researchers could include device-specific features (such as device type, manufacturer, and software version), user behaviour features (such as frequency and timing of device usage), and environmental features (such as network conditions). These additional features could provide a more comprehensive view of the IoMT environment and potentially improve the performance of the intrusion detection system.

4 CONCLUSION

We have contributed to knowledge in this field by building an anomaly detection system for the Internet of Medical Things based on Artificial Intelligence techniques using Machine and Deep Learning models. We implemented six ML and DL algorithms: RF, SVC, K-NN, CNN, hybrid CNN-LSTM, and the attention-based hybrid CNN-LSTM for IDS for IoMT. We performed a hyperparameter optimisation using the Random search algorithm to obtain the best parameters needed to build our models in this study. State-of-the-art ML and DL architectures are built in this study to investigate the best performance in terms of selected performance metrics in the task of detecting anomalies in IoMT networks. To train our models and develop an efficient detection system, we adopted the popular AIDS dataset that is representative of the current attacks on IoMT devices and networks.

We also selected the SMOTE oversampling technique to handle the class imbalance in our training dataset. Our experiment results show that in general the RF and the KNN models are best performed in terms of all metrics. The attention-based hybrid CNN-LSTM model is second best, while the hybrid CNN-LSTM model is worst performed. In terms of classifying all attack labels by the different metrics, we found that no particular model performed best, as different models performed differently in terms of different metrics while classifying the cyber-attack labels. However, we acknowledge the limitations of our study, including the use of a single dataset and the variations in model performance across different attack labels and metrics. These limitations highlight the need for future research to incorporate multiple, diverse datasets and to consider multiple evaluation metrics to enhance the generalizability and reliability of the proposed system. We believe that addressing these limitations in future research endeavours will further advance the field of IoMT security, leading to more robust and reliable intrusion detection systems.

AUTHOR'S CONTRIBUTION

Conceptualization, B.P.; methodology, E.F. and B.P.; software, experiment E.F; writing E.F and B.P.; supervision, B.P.

COMPETING INTERESTS

All authors declare that the paper is free of a conflict of interests (COI) or competing interests (CI).

REFERENCES

[1] R. Dwivedi, D. Mehrotra, and S. Chandra, "Potential of Internet of Medical Things (IoMT) applications in building a smart healthcare system: A systematic review," *Journal of Oral Biology and Craniofacial Research*, vol. 12, no. 2, p. 302, Dec. 2021, doi: 10.1016/j.jobcr.2021.11.010.

[2] "Full article: Internet of Medical Things (IoMT): Overview, Emerging Technologies, and Case Studies." Accessed: Nov. 16, 2024. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/02564602.2021.1927863

[3] "Internet of Medical Things (IoMT) - An overview | IEEE Conference Publication | IEEE Xplore." Accessed: Nov. 16, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/9075733

[4] J. Srivastava, S. Routray, S. Ahmad, and M. M. Waris, "Internet of Medical Things (IoMT)-Based Smart Healthcare System: Trends and Progress," *Computational Intelligence and Neuroscience*, vol. 2022, p. 7218113, Jul. 2022, doi: 10.1155/2022/7218113.

[5] Liyakathunisa, A. Alsaeedi, S. Jabeen, and H. Kolivand, "Ambient assisted living framework for elderly care using Internet of medical things, smart sensors, and GRU deep learning techniques," *Journal of Ambient Intelligence and Smart Environments*, vol. 14, no. 1, pp. 5–23, Jan. 2022, doi: 10.3233/AIS-210162.

[6] B. Pranggono, K. McLaughlin, Y. Yang, and S. Sezer, "Intrusion Detection Systems for Critical Infrastructure," in *The State of the Art in Intrusion Prevention and Detection*, Al-Sakib Khan Pathan, Ed., CRC Press, 2014, pp. 115–138.

[7] P. Mishra, V. Varadharajan, U. Tupakula, and E. S. Pilli, "A Detailed Investigation and Analysis of Using Machine Learning Techniques for Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 686–728, 2019, doi: 10.1109/COMST.2018.2847722.

[8] "Anomaly-Based Detection - an overview | ScienceDirect Topics." Accessed: Nov. 16, 2024. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/anomaly-based-detection

[9] B. Pranggono and A. Arabo, "COVID-19 pandemic cybersecurity issues," *Internet Technology Letters*, vol. 4, no. 2, p. e247, 2021.

[10] A. Si-Ahmed, M. A. Al-Garadi, and N. Boustia, "Survey of Machine Learning Based Intrusion Detection Methods for Internet of Medical Things," Mar. 07, 2023, *arXiv*: arXiv:2202.09657. Accessed: Nov. 16, 2024. [Online]. Available: http://arxiv.org/abs/2202.09657

[11] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 1, p. 18, Mar. 2021, doi: 10.1186/s42400-021-00077-7.

[12] A. Binbusayyis, H. Alaskar, T. Vaiyapuri, and M. Dinesh, "An investigation and comparison of machine learning approaches for intrusion detection in IoMT network," *J Supercomput*, vol. 78, no. 15, pp. 17403–17422, Oct. 2022, doi: 10.1007/s11227-022-04568-3.

[13] G. Thamilarasu, A. Odesile, and A. Hoang, "An Intrusion Detection System for Internet of Medical Things," *IEEE Access*, vol. 8, pp. 181560–181576, 2020, doi: 10.1109/ACCESS.2020.3026260.

[14] G. Zachos, I. Essop, G. Mantas, K. Porfyrakis, J. C. Ribeiro, and J. Rodriguez, "An Anomaly-Based Intrusion Detection System for Internet of Medical Things Networks," *Electronics*, vol. 10, no. 21, Art. no. 21, Jan. 2021, doi: 10.3390/electronics10212562.

[15] J. Wang, H. Jin, J. Chen, J. Tan, and K. Zhong, "Anomaly detection in Internet of medical Things with Blockchain from the perspective of deep neural network," *Information Sciences*, vol. 617, pp. 133–149, Dec. 2022, doi: 10.1016/j.ins.2022.10.060.

[16] S. Bacha, A. Aljuhani, K. B. Abdellafou, O. Taouali, N. Liouane, and M. Alazab, "Anomaly-based intrusion detection system in IoT using kernel extreme learning machine," *J Ambient Intel Human Comput*, May 2022, doi: 10.1007/s12652-022-03887-w.

[17] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, "TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems," *IEEE Access*, vol. 8, pp. 165130–165150, 2020, doi: 10.1109/ACCESS.2020.3022862.

[18] N. Moustafa, M. Keshky, E. Debiez, and H. Janicke, "Federated TON_IoT Windows Datasets for Evaluating AI-Based Security Applications," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Guangzhou, China: IEEE, Dec. 2020, pp. 848–855. doi: 10.1109/TrustCom50675.2020.00114.

[19] N. Moustafa, M. Ahmed, and S. Ahmed, "Data Analytics-Enabled Intrusion Detection: Evaluations of ToN_IoT Linux Datasets," in *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Guangzhou, China: IEEE, Dec. 2020, pp. 727–735. doi: 10.1109/TrustCom50675.2020.00100.

[20] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

[21] "SMOTE: synthetic minority over-sampling technique: Journal of Artificial Intelligence Research: Vol 16, No 1." Accessed: Nov. 16, 2024. [Online]. Available: https://dl.acm.org/doi/10.5555/1622407.1622416

[22] "A Comprehensive Analysis of Synthetic Minority Oversampling Technique (SMOTE) for handling class imbalance - ScienceDirect." Accessed: Nov. 16, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0020025519306838

[23] "Normalization (with MinMaxScaler)." Accessed: Nov. 16, 2024. [Online]. Available: https://kaggle.com/code/kiranvairagade/normalization-with-minmaxscaler

[24] M. U. Siregar, I. Setiawan, N. Z. Akmal, D. Wardani, Y. Yunitasari, and A. Wijayanto, "Optimized Random Forest Classifier Basedon Genetic Algorithm for Heart Failure Prediction," in *2022 Seventh International Conference on Informatics and Computing (ICIC)*, Dec. 2022, pp. 01–06. doi: 10.1109/ICIC56845.2022.10006987.

[25] N. Jannah, S. Hadjiloucas, and J. Al-Malki, "Arrhythmia detection using multi-lead ECG spectra and Complex Support Vector Machine Classifiers," *Procedia Computer Science*, vol. 194, pp. 69–79, Jan. 2021, doi: 10.1016/j.procs.2021.10.060.

[26] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. null, pp. 281–305, Feb. 2012.

[27] M. M. Forootan, I. Larki, R. Zahedi, and A. Ahmadi, "Machine Learning and Deep Learning in Energy Systems: A Review," *Sustainability*, vol. 14, no. 8, Art. no. 8, Jan. 2022, doi: 10.3390/su14084832.

[28] P. Madan, V. Singh, D. P. Singh, M. Diwakar, B. Pant, and A. Kishor, "A Hybrid Deep Learning Approach for ECG-Based Arrhythmia Classification," *Bioengineering*, vol. 9, no. 4, Art. no. 4, Apr. 2022, doi: 10.3390/bioengineering9040152.

[29] D. Bahdanau, K. Cho, and Y. Bengio, "Neural Machine Translation by Jointly Learning to Align and Translate," May 19, 2016, *arXiv*: arXiv:1409.0473. doi: 10.48550/arXiv.1409.0473.

[30] M.-T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," Sep. 20, 2015, *arXiv*: arXiv:1508.04025. doi: 10.48550/arXiv.1508.04025.